BITS,BYTES&PIXELS

LIMA 99/4A USERS GROUP

February 1988 Volume 4 #2

PLATO CATALOG XB ROUTINE DOCUMENTATION

by Mike Richardson
Lima Ohio User Group

GENERAL COMMENTS: Trying to catalog a Plato disk by conventional techniques is futile since the file descriptor index is at sector >0167. You can copy this sector to sector >0001, and then catalog the disk. However, the file names returned are not very meaningful. (If anyone knows what the names mean I would appreciate a note.) I wrote an a/l routine that actually catalogs the Plato disk by reading the information in the TIMENU and DISKMENU files. To use my routine you need extended basic, 32K, and a disk system with TWO drives or one drive and a ramdisk.

LOADING: This program is a combination of XBASIC and assembly language, and loads directly from XBASIC with OLD DSK1.PLATO. Before you RUN the program, you need to put an initialized disk in Drive #2. The program expects to find the Plato disk in drive #1 and you are prompted to insert the PLATO disk. Press <ENTER> rather than "N" at the flashing "N" prompt. Don't attempt to edit the XBASIC protion of the program. If you do, things probably won't work properly. Once loaded, you can invoke the routine from XB command mode with Call Link("PLATO",TIMENU$(),TI,DISKMENU$(),DSK). After invoking the routine the TIMENU$ string array contains information on the curriculum, subject, and program while the TI numeric variable tells us how many strings were passed back in the string array. Likewise the DISKMENU$ array has information on the specific titles of the Plato programs and DSK specifies the number of string variables passed back.

PROGRAM EXECUTION: The program creates a file D/VB0 file called PLATOCAT on the data disk in DSK2. You are asked to give each plato disk your own code name. You might, for example, give them consecutive numbers. For each disk, PLATOCAT gives your special name, CURRICULUM name (such as High School Skills), SUBJECT name (such as Biology), PROGRAM PACKAGE (such as Biology 4), and the names of the files that appear on the disk's table of contents (such as Health tutorial, Health drill). PLATOCAT is added to as you catalog subsequent PLATO disks, and can be read and printed with Funnelweb or TI Writer. This plato catalog routine does not itself read PLATOCAT to the screen. Remember to press <ENTER> every time the flashing "N" prompt appears.

ERRORS: If the a/l routine encounters an error, it traps the error code and passes the code back to XB.

LIMITATIONS: Not all Plato disk have TIMENU and DISKMENU files. Consequently this program cannot catalog these disk. It will seach all 359 sectors looking for these file names. This was a compromise on my part as I didn't feel it was necessary to step the drive heads out to sector >0167 to read the file descriptor index each and every time.

ACKNOWLEGEMENTS: I used Barry Boones Systex program to hybridize the a/l routine and a DSRLNK routine from John Clulow.

Mike Richardson 12/12/87
927 N. Canal St.
Delphos, Ohio 45833
216-456-0478

EDITOR'S NOTE: User Groups (not individuals) may obtain a copy of this program by sending a disk and mailer to the Lima User Group, P.O. Box 647, Venedocia OH 45894.

Below is a sample printout of a small PLATOCAT file.

```
1
CURRICULUM:
High School Skills
Science
SUBJECT:
Biology
PROGRAM PACKAGE:
Biology 4
Health
Tutorial
Health
Drill
2
CURRICULUM:
High School Skills
Science
SUBJECT:
Biology
PROGRAM PACKAGE:
Biology 4
Population and
Environment
Tutorial
Population and
Environment
Drill
```

DONE

COMMENTS ON *CERTIFICATE 99*

by Charles Good
Lima Ohio User Group

A review of this software was promised in the last issue of BB&P. Since then, a rather comprehensive review by Steven B. Mehr has appeared in the Dec. 87 issue of Micropendium. What follows is not a complete description of Certificate 99, but rather some additional comments and documentation aids.

I really think Certificate 99 deserves better report card grades than those assigned by Mr. Mehr. It is VERY easy to use, and most people can follow the screen prompts successfully the very first time without referring to the documentation. It produces excellent certificates and signs. We have all seen "Print Shop" type signs printed with a border on a vertically alligned piece of 8.5 x 11 inch paper. Certificate 99 will do just as good a job making such signs, only the signs are alligned horizontally. This ability to print quick professional looking signs is perhaps the most useful potential for Certificate 99. This is truely application software. With it, your computer and printer can do something really useful.

I purchased my copy from TexComp. It came in a very nice transparent plastic envelope with disk, documentation, and 6 single sheets of fancy parchment-like paper. I also received, outside the envelope 30 fan-fold tractor feed sheets of such paper. With the tractor feed perforations this paper would be too wide to fit into the envelope. I am not sure, but I suspect that this was a "free" bonus only from TexComp. You can get more of this fancy tractor feed paper from TexComp for $9.95/250 sheets.

The documentation carries a very strong copy protection notice. The name and address of the owner are said to be encoded on the protected disk. I know that when this software was sold at the recent Chicago TI Faire the authors/vendors (one and the same) took each disk sold and appeared to encode something on it prior to turning it over to the purchaser. I really doubt, however, if a mass merchandiser such as TexComp or Triton would go to this kind of trouble. They probably just take the package off the shelf and put it in the mail. The original disk is said to be copy protected, and Great Lakes Software specifically does NOT offer to sell cheap backup disks. Since you are supposed to use both sides of the disk with the write protect tabs OFF, I felt very insecure about writing data to my one and only original disk. I therefore put write protect tabs over the holes for both sides of my original flippy disk and made my legal back up copy using DM1000's "copy disk" function. I had no trouble at all backing up my original this way, and use only my backup for actual certificate or sign creation.

All of Mr. Mehr's comments about test entry are true, but text entry is really quite easy. You have full screen editing using FCTN and the arrow keys. Pressing <ENTER> drops you down to the left edge of the next lowest line. This may cause the cursor to disappear, however, because in the small sized letter field the left column is at the very left edge of the 99/4A's screen display. TVs and even some monitors don't show this extreme left column on the screen. The first time I used Certificate 99 on a friend's system (my monitor shows the cursor all the time) I was very mystified by the disappearing cursor.

The DOCs state that you can alter the choice of graphics with most of the standard "artist" programs. The two files of graphics are 25 sector PROGRAM files (CDATA21 and CDATA22), and each of these files contains 12 graphics on its single screen. If you use another 25 sector PROGRAM graphic, be aware that only 1/12 of the graphic at a time can be used by CERTIFICATE 99 to ornament a sign or certificate. The original CDATA21 22 files are reproduced here graphically for reference. In my opinion such reproductions should have been included in the DOCs.

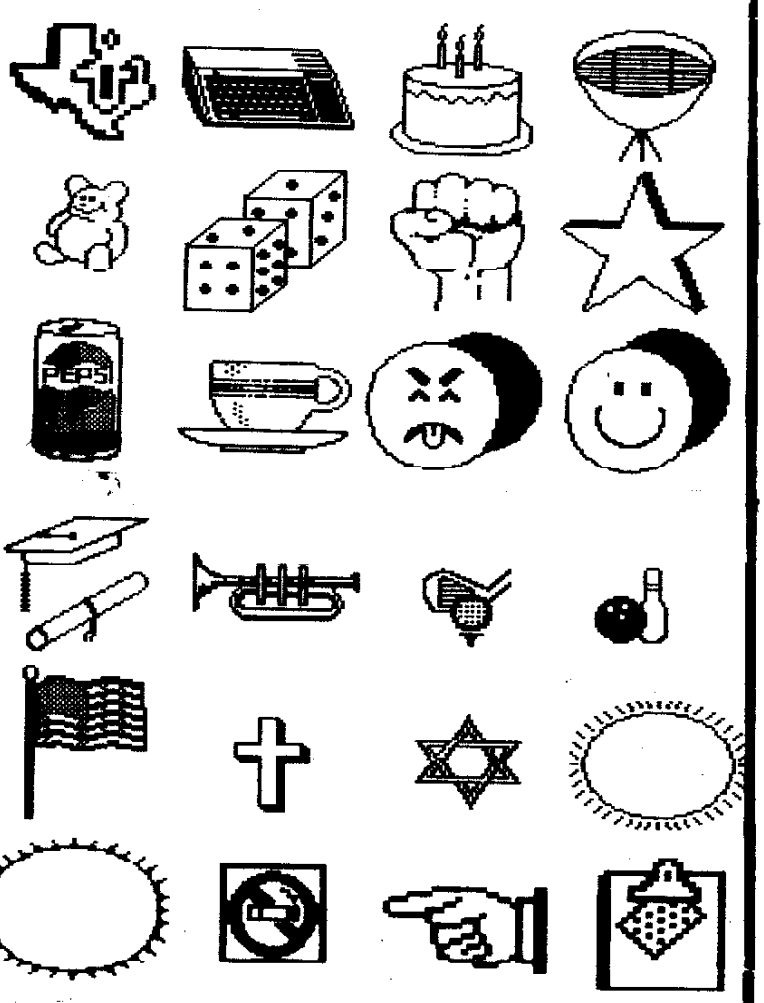
An important UNDOCUMENTED feature is the ability to save files of defaults for a particular certificate or sign (font, text, graphic, border, signature, screen colors) and to load them onto the disk prior to using CERTIFICATE 99. The program normally saves all the data from the last certificate as the defaults for the next certificate. These data are saved as file CDATA00 on side 2 of the disk. There is also a CDATA00 file on side one, but I don't know what this file does. You can create a collection of certificate files by using a disk manager to lift the side two CDATA00 files off the system disk and keep these files on a data disk, each with a different file name. To print a particular previously created certificate use a disk manager to put the data file back onto side two of the system disk, rename the file CDATA00, and then boot the program. This procedure is, of course, cumbersome, but at least such a procedure exists. I don't understand why this feature isn't mentioned in the DOCs. It would be much nicer if you were prompted from within the program for the file name of a previously created certificate.

Neither the DOCs, nor the several reviews I have read specifically state the default color combinations or available signatures. These are listed here in the accompanying reference sheet. Richard A. Paquette is listed among the famous people whose signatures are available. (You can also choose no signature, or a blank line for your own signature.) He is the CERTIFICATE-99 author, in case you are wondering. Since it is not possible to cycle through the various screen color combinations, as you can with the graphics and borders, these color combinations should have been listed in the DOCs.

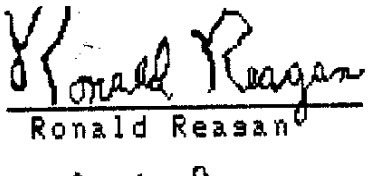
NEXT PAGE

CERTIFICATE-99 REFERENCE SHEET

AVAILABLE GRAPHIC



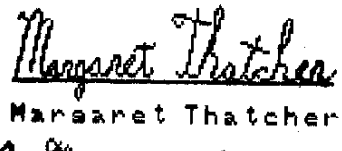
AVAILABLE SIGNATURES

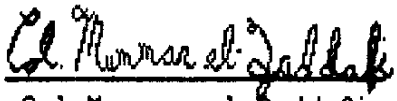

Ronald Reagan

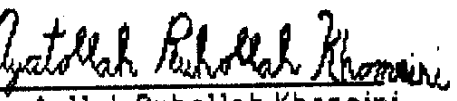
Lee A. Iacocca

Santa Claus

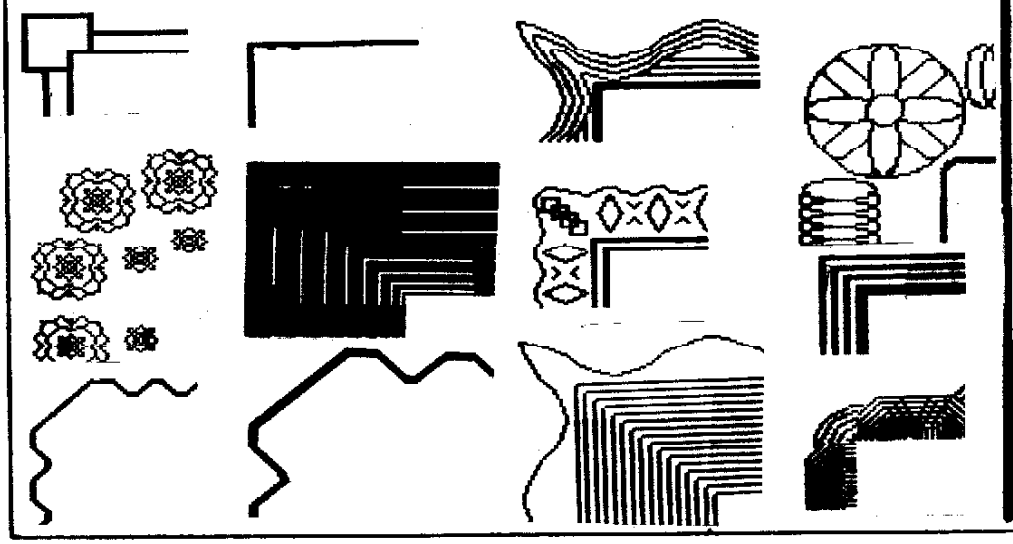
Richard A. Paquette


Margaret Thatcher


Col. Mummar el-Qaddafi


Ayatollah Ruhollah Khomeini

AVAILABLE BORDERS



SCREEN COLOR COMBINATIONS

1. Black on Green
2. White on Lt. Blue
3. White on Black
4. Black on Purple

DONE

ED. NOTE: The following article describes a SOFTWARE EMULATION OF *SUPER EXTENDED BASIC (v120)* using the regular extended basic module. It was sent to us by the author Art Byers, and was first published in the Jan. 88 issue of CALL SOUNDS, the newsletter of the Central Westchester user group. The entire collection of sub programs is in the public domain and can be downloaded from Delphi. -

The BIT BUCKET #7, CALL SOUNDS, 1/1/88 by Art Byers, Sysop Delphi TINET

At the November '87 meeting of the Central Westchester 99'ers, Bob Cataldo gave us a very fine demonstration of the new module Super Extended Basic. I own three regular Extended BAsic modules and am not about to spend for a 4th, even tho it does have many nice features. Besides practically every "advanced " feature has been kicking around the 99/4A community for years, either as an assembly routine or as a CALLed SUB program in TI Extended Basic.

I do not know if it was Barry Traver or Jim Peterson who first published the GOTO(n), GOSUB(n), RESTORE(n), and RUNPROG("device.filename") [alternate RUNPROG(F$)]. Thanks to BOTH!!!!! The Smart Programer and the Genial Traveler have also published PEEKVdp and POKEVdp sub programs. A fine "CLOCK" Subprogram for Lo Memory was published by the Smart Programmer, and Barry's GT recently published a one line program that enables you to circumvent the auto LOAD boot of XB - but that too has been around for years!

Furthermore, in many many years of programming in TI XB, I have never really had the need to use GOSUB(n), RESTORE(n) or GOTO(n) and considered those subprograms interesting but not vital. The RUNPROG(F$) -IS- a very important feature.

The end result is that at the January meeting of the CW 99'ers, a disk was given out that will enable members to have a high degree of compatability with the new Super Extended Basic, using only their regular XB Module. In addition, any member of this club that finds a SUPER XB Module program that will not run using the merge SUB programs on this disk can brink me a copy and I will rewrite it so as they CAN use it. Obviously, I cannot afford to buy the QUALITY SOFT "Draw 'n Plot" disk for all the club. If they want more complete compatability, they will have to buy that for themselves.

I am still searching for the proper CALL LOADs and CALL PEEKs that will enable me to duplicate a few subprograms in the Super XB Module. The subprograms not included below because of this are: SUB CLKOFF, SUB ALOCK, SUB CTRL, SUB FCTN,SUB SCROFF, and SUB SCRON

Here is the documentation for the disk, plus a printout of most of the CALL SUB programs:

SUPERXB#V1 - Documentation.

This disk contains a series of TI Extended Basic CALL SUBprograms that attempt to make the TI Extended Basic module mostly compatable with the new Super Extended Basic Module.

The New Super Extended Basic module will load QUALITYSOFT's Draw 'n Plot routines into low memory. To obtain this to use with the TI XB module, you must buy them from Quality Soft and preload them into lo memory. Complete documentation and instructions come with the purchase.

The PEEKVdp ram and POKEVdp ram routines are built into the new Super XB module. The only way you can use these CALLs with the regular XB module is to preload assembly language routines, such as those that appeared in the SMART PROGRAMMER or those available by subscribing to the GENIAL TRAVELER. These routines will reside in Lo-memory and be accessed by CALL LINKs. You may have to rewrite the SUB PEEKV and SUB POKEV subprogrtrams to conform with the requirements of those assembly programs. Also, they will cannot be used if "Draw 'n Plot" is in lo-memory, as they will over-write those sub programs.

Similarly, the clock programs are in the Super XB module. To simulate this, you will have to preload an assembly language clock routine such as the one tat appeared in the August 1984 Issue of the Smart programmer. Again, if the Draw on Plot subprograms are in low memory, these will over write them and probably cause lockup.

What is obvious from the above is that you cannot be 100% compatable with a program written for the new Super XB module if it makes use of all the above calls in the same program. However, most programs do not use all these subprograms. In addition, the SUB programs provided are all very useful and can be used in your own regular XB programs.

The SUB programs are provided in MERGE form, consecutively numbered. MERGE in only those you need for the program you wish to run or ar planning to write. See the TI Extended Basic reference Guide for the proper way to use CALL SUB programs.

This disk contains the following CALLs which are 100% compatable with the CALLs built into the new Super XB module:

CALL ALL(numeric expression) This call fills the screen with the character of the ASCII number in the perenthesis. ie: CALL ALL(32) fills the screen with blanks, just like CALL CLEAR. CALL ALL(36) fills the screen with dollar signs -$-.

CALL BEEP sounds a beep tone.

CALL BYE is the same as the immediate command BYE except that it is used in a program.

NEXT PAGE

Bits, Bytes & Pixels

CALL CHIMES sounds a single chime. Although it is possible to duplicate the chimes provided in the ditor Assembler Manual, I have chosen to use a chime from one of Jim Peterson's Tips from the TIGERCUB. Jim is the undisputed KING of XB CALL SUB programs, and this was included as a tribute to him. You can buy over 300 usefull CALL subs in merge form on his three Nuts and Bolts disks - and get a real education on how to program in XB.

CALL CLOCK will only work if you have the clock routine preloaded into Low memory.

CALL COLORS(foreground,background). This call sets the foreground and background colors for all Character sets except set 0. Therefore it does not set the border colors. These can be done by the standard CALL SCREEN(). By setting the background color to TRANSPARENT -1- in CALL COLORS, the color called by CALL SCREEN() will become the background color.

CALL GOSPRT starts all sprites moving after they have been stopped with CALL STSPRT.

WARNING: only ONE of the following can be used in any program as the last line of the CALL SUB XXXX(n) absolutely MUST be the last line of the program. Therefore, they all use the same line numbers, ending in 32767, and will overwrite each other: GOSUB(n), GOTO(n), RESTORE(n), RUNPROG("device,filename"). CALL GOSUB(n) allows you to use a variable with GOSUB. However, remember when you RESequence line numbers, (n) will NOT be resequenced. Also remember that a call to a non existing line number will crash your program. CALL GOTO(n) same as above but is a GOTO instead of a GOSUB. CALLL RESTORE(n) allows the sue of a variable instead of a fixed line number with the same cautions as in GOSUB(n). CALL RUNPROG("device.filename") circumvents TI XB's disallowance of RUN "DSK1."A$" or RUN A$ where A$ might be ___ "DSK3.MYPROGRAM"

CALL HONK sounds a warning honk.

CALL KEYS("keylist",numeric variable) allows validation of the keys listed by either a predefined string (A$) or listed in between the quotes ie:"1234ABC" and returns a numberic variable. If the key A was pressed the variable would be 5 as A is the fifth in the sequence shown. If a key other than those defined is pressed, a honk is sounded and the program awaits a correct key press. IMPORTANT: CALL HONK must be MERGED along with SUB KEYS(). I consider this to be among the most usefull of all the subprograms on this disk.

CALL NEW is the same as the immediate command NEW but can be used in a program.

CALL PEEKV(vdp address, value list) and CALL POKEV(vdp address, value list) are explained above.

CALL QUITOFF disables the quit key, FCTN=

CALL QUITON enables the quit key, FCTN=

CALL STSPRT stops all sprite motion. To restart use CALL GOSPRT.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  SUB PROGRAMS and documentation
written and donated to the public domain
by ART BYERS. SYSOP TINET special
interest group on DELPHI. 12/16/87.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

30000 SUB ALL(N)
30001 CALL VCHAR(1,1,N,768):: SUBEND

30002 SUB BEEP
30003 CALL SOUND(220,880,1):: SUBEND

30004 SUB BYE
30005 CALL LOAD(-31962,32):: SUBEND

30006 SUB CHIMES
30007 FOR N=0 TO 25 STEP 5 :: CALL
SOUND(-999,1047,N,784,N,659,N):: NEXT N
:: FOR N=26 TO 30 :: CALL
SOUND(-999,1047,N,784,N,659,N):: NEXT N
:: SUBEND

30008 SUB COLORS(F,B)
30009 FOR N=1 TO 14 :: CALL
COLOR(N,F,B):: NEXT N :: SUBEND

30010 SUB GOSPRT
30011 CALL LOAD(-31806,0):: SUBEND

30012 SUB HONK
30013 CALL SOUND(220,220,1):: SUBEND
30014 SUB KEYS(A$,P)
30015 CALL KEY(0,K,S):: IF S=0 THEN
30015
30016 P=POS(A$,CHR$(K),1):: IF P>0 THEN
30017 ELSE CALL HONK :: GOTO 30015
30017 SUBEND

30018 SUB NEW
30019 CALL LOAD(-31952,255,231,
255,231):: SUBEND

30020 SUB QUITOFF
30021 CALL LOAD(-31806,16):: SUBEND

30022 SUB QUITON
30023 CALL LOAD(-31806,0):: SUBEND
```

NEXT PAGE

```
30024 SUB STSPRT
30025 CALL LOAD(-31806,64):: SUBEND


30026 SUB CLOCK
30027 CALL CLEAR :: CALL LINK("CLOCK")
30028 DISPLAY AT(12,1):"TIME?000000" ::
ACCEPT AT(12,6)SIZE(-6)BEEP
VALIDATE(DIG IT):TIME$ :: CALL
LINK("SETCLK",TIME$):: SUBEND

32762 SUB GOSUB(N)
32763 CALL PEEK(-31952,I,J):: CALL
PEEK(I*256+J-65534,I,J)
32764 DEF L1(N)=INT(N/256)
32765 DEF L2(N)=N-256*(INT(N/256))
32766 N=1000*N :: CALL
LOAD(I*256+J-65534,L1(N),L2(N))
32767 GOSUB 12345 :: SUBEND

32762 SUB GOTO(N)
32763 CALL PEEK(-31952,I,J):: CALL
PEEK(I*256+J-65534,I,J)
32764 DEF L1(N)=INT(N/256)
32765 DEF L2(N)=N-256*(INT(N/256))
32766 N=1000*N :: CALL
LOAD(I*256+J-65534,L1(N),L2(N))
32767 GOTO 12345 :: SUBEND

32762 SUB RESTORE(N)
32763 CALL PEEK(-31952,I,J):: CALL
PEEK(I*256+J-65534,I,J)
32764 DEF L1(N)=INT(N/256)
32765 DEF L2(N)=N-256*(INT(N/256))
32766 N=1000*N :: CALL
LOAD(I*256+J-65534,L1(N),L2(N))
32767 RESTORE 12345 :: SUBEND

32762 SUB RUNPROG(FN$)
32763 REM THANKS TO BARRY TRAVER
32764 DISPLAY AT(24,1)ERASE ALL:"
Loading ";FN$
32765 CALL PEEK(-31952,I,J):: CALL
PEEK(I*256+J-65534,I,J)::
@=I*256+J-65534 :: D$=FN$ :: CALL
LOAD(@,LEN(D$))
32766 FOR I=1 TO LEN(D$):: CALL
LOAD(@+I,ASC(SEG$(D$,I,1))):: NEXT I ::
CALL LOAD(@+I,0)
32767 RUN "DSK1.FILENAME" :: SUBEND
```

****

****DONE****

32K IN THE CONSOLE

by Melvin Nomina
Lima Ohio User Group

As I had some time off thru the holidays I decided to try Mike Ballmann's 32K addition to the 16 bit bus in the kids console. It was easy for me, and only took about 2 hours. The accompanying article by John Clulow, which originally appeared in the July 87 issue of Northwest Ohio 99er News, tells you how. I have included here the Logic Board Component Location Diagram from the 99/4A Technical Data Book and my own circuit diagram to aid you in this project.

All needed parts can be purchased as a kit from
Bud Mills Services
116 Dartmouth Drive
Toledo OH 43614
The kit costs $28, your total cost is about $30.

There are some advanages to having 32k in the console! How many times did you want to get on the computer and the kids have yours tied up playing games? If you don't have a 32K in your PE box, or your kids' system lacks memory expansion this would be an inexpensive way to go. This internal 32K isn't compatible with any other 32K in your system. You can, however run any EA#5 program from tape. Such programs can be transferred from disk to tape with the tape loader published in the June 87 issue of BB&P or with the public domain software Disk Information Manager.

In addition to the 32K, the other only thing you need to load EA#5 software once it is on tape is the Editor Assembler command module. If you don't already have one you can make one up by ordering the EA module ROM chip, part #101-5960-1204 from TI. Call TI at 800-842-2737 to place your order. The cost is about $2. You can install the chip in one of your old game modules.

Good luck! As is usual with such projects, proceed at your own risk. If you arn't handy with a soldering iron, get someone else to do this poject for you.

MIKE BALLMANN'S 32K -- 16 BIT BUSS PROJECT

by John Clulow

The following is a step by step description of how to add 64K of RAM memory on the 16 bit bus. The present modification uses only 32K. This corresponds to the memory space of the 32K Memory Expansion. The modification yields a speed increase of about 50%.

Mike Ballmann is currently working on a circuit to allow CRU decoding of the remaining 32K. This will open up a whole new area of software, including such possibilities as a real DOS which could be loaded into RAM from disk on power-up. The 32K modification described below can easily be modified for full decoding upon completion of Mike's work.

NEXT PAGE

You will need two Hitachi HM62256LP-12 RAMs. One source of these is Microprocesors Unlimited. They cost around $12. You'll also need a 74LS21 and a 74LS153. These can be obtained from various electronics supply houses. All wiring should be done with wire-wrap wire. You should use a low wattage soldering pencil with a fine pencil type tip.

The modification is done on the main board of the Black Silver console, and you'll need to refer to the (accompanying) Logic Board Component Location Diagram from the 99/4A console Technical Data book.

1. Remove the board from the console and identify the two ROMs. They are located between the GROM connector and the 9900 IC. One is parallel to the 9900 and the other is perpendicular to it. They are U610 an U611 on the Component Location Diagram.

2. Bend the pins on the HM62256 IC's closer so they will firmly contact the ROM pins when piggy-backed. One way of doing this is to place the RAM in it's side on a table and then move the body of the IC toward the table to bend the pins uniformly.

3. Bend out the following pins on both HM62256 RAMs: 1 2 20 22 23 26 27 28. These pins will NOT be soldered to anything on the ROMs. Holding the IC with the notch up and looking at the top, pin numbers start with pin 1 on the upper left, go down the left side, then across the and up the right side. Pin 28 is opposite pin 1 on the end with the notch.

4. Place one HM62256 over the ROM that is parallel to the 9900. Make sure th notch points toward the 9900 and that the writing on the 9900 and the 62256 can be read from the same direction. Place the RAM such that pins 1 2 27 and 28 extend beyond the end of the ROM. The un-notched end of the RAM should line up with the un-notched end of the ROM. There should be a sort of "spring tension" that clamps the RAM pins onto corresponding ROM pins below it. This will help to insure good solder joints. If the RAM doesn't fit tightly, remove it and bend the pins closer.

5. Solder all RAM pins not bent out to the ROM pins below. Use a low wattzge pencil with a fine pencil type tip. Inspect each solder joint carefully in good light under magnification.

6. Place the second 62256 on the ROM that is perpendicular to the 9900. The notch on the RAM points away from the 9900 and toward the edge of the board. As above, solder and inspect all pins that were not bent out.

7. Bend out the 74LS21 pins 1 2 4 5 6 8 10 12 14. Note that pins 1 and 14 are across from each other on this 14 pin IC.

8. The 74LS21 will be piggy-backed on the 74LS138 U504. this IC is located adjacent to the end of the board where the edge connector is. There are two 138's next to each other. U504 is the one nearest the end of the board. You will place the 74LS21 so that the UN-NOTCHED end lines up with the un-noched end of the 138 (pointing toward the cassette connector). Pins 1 and 16 of the 138 will extend beyond the notched end of the 74LS21.

9. Before positioning the 74LS21, solder 1/2" lengths of wire-wrap wire to the 138 pins 7 and 9. Then position the 74LS21 on top of the 138 and solder all pins not bent out to the 138 pins below and inspect the connections.

10. Bend out all of the 74LS153 pins EXCEPT 8 and 16.

11. Place the 153 over U613, a 74LS194. The notch will line up with the 194 notch and point toward the edge of the board away from the 9900. Solder pins 8 and 16 of the 153 to pins 8 and 16 of the 194 below.

12. At the end of the 9900 opposite to where the RAM's have been piggy-backed, you will see a line of three ICs. They are 74LS00, 74LS32, and 74LS04. The 74LS00 is U606 and the 74LS32 is U605. Turn the board upside down so you can see the traces. Find the trace that runs from pin 11 of the 74LS00 (U606) to pin 13 of the 74LS32 (U605). Double check to make sure you're doing the pin numbering correctly. When you've found the trace, cut it with a knife so there is no continuity between the LS00 pin 11 and the LS32 pin 13.

13. Identify the piggy-backed RAM that is perpendicular to the 9900. Solder wire-wrap wires connecting every bent out pin on this RAM to the corresponding bent out pin on the RAM that is parallel to the 9900. Pin 1 to pin 1, pin 2 to pin 2, etc. There will be eight wires in all to solder.

14. Solder wire-wrap wires to make the following connections on the RAM that is parallel to the 9900. Pin 1 goes to pin 24 of the 9900 (solder the wire to the 9900 pin on top of the board). Pin 2 goes to the 9900 pin 22. Pin 20 goes to two places. Connect pin 20 of the RAM to pin 22 of the RAM and also to pin 8 (bent out) of the 74LS21. There should be three wires comming off pin 20 of the RAM. Pin 23 of the RAM goes to pin 21 of the 9900. Pin 26 of the RAM goes to 23 of the 9900. Pin 27 of the RAM goes to pin 61 of th 9900 (fourth from the top on the right side). Finally, connect pin 28 of the RAM to pin 20 of the 74LS244 (UJ10) adjacent ton the piggy-backed 74LS21.

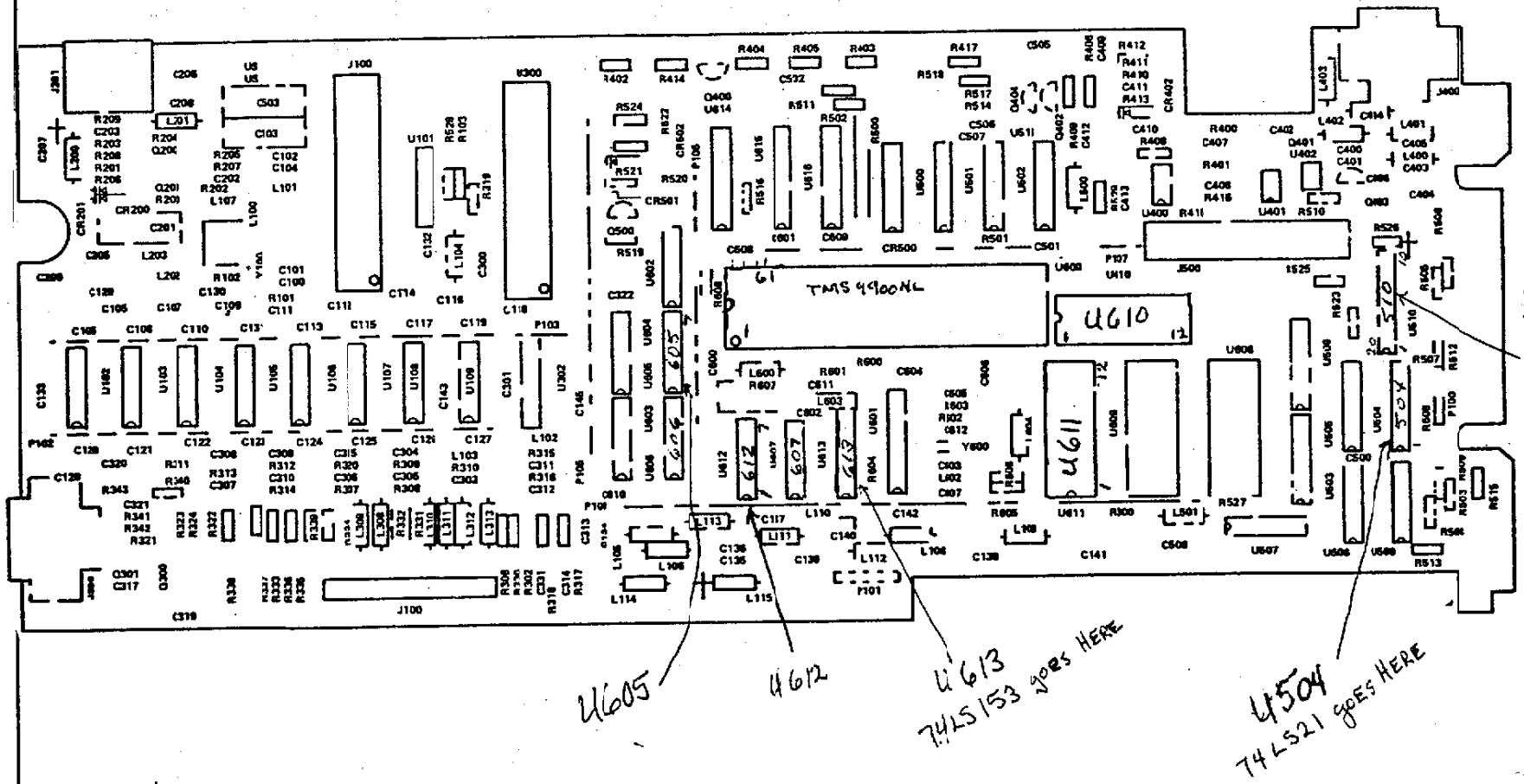
15. Connect the following 74LS21 pins with a bare wire: 1 2 4 and 14. Connect the short wire from the 138 pin 7 to the LS21 pin 5 (bent out). Connect the LS21 pin 6 to LS21 pin 12. Connect LS21 pin 8 (bent out) to the piggy-backed 153 pin 2. Connect the short wire comming from the 138 pin 9 to LS21 pin 10. Finally, connect the 74LS21 pin 14 to the 74LS244 pin 20 that you connected the RAM pin 28 to.

16. OK, we're almost done, so take a break and have a ~~beer~~ coke.

17. On the 153, connect pin 9 to pin 13 on the 74LS32 (U605). Pin 10 of the 153 goes to pin 14 of the 74LS74 next to it (U607). Also connect pin 10 of the 153 to pin 15 of the 153, and then connect pin 15 of the 153 to pin 7 of the 74LS00 U612 (next to the 74LS74). Connect pin 14 of the 153 to pin 11 of the 74LS00 U606; that's the one you cut the trace on.

That's it! Now have another ~~beer~~ coke before putting your computer back together. When you try it out, remember that this version isn't compatable with any other 32K in the system. If you have any problems with this I can't promise I can help, but feel free to give me a call or write EMAIL 419-874-8838. Ask for John (or Hose-Head).

NEXT PAGE



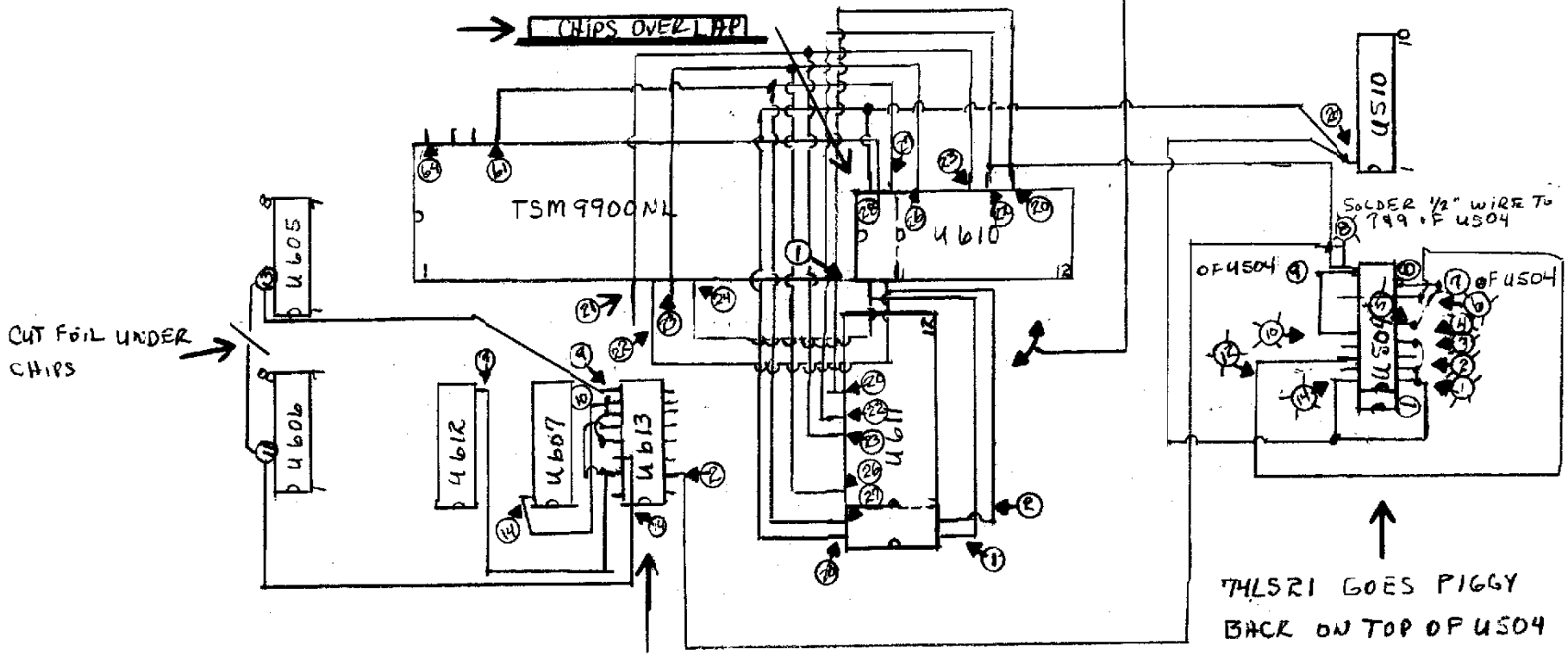
TI-99/4A LOGIC BOARD COMPONENT LOCATION DIAGRAM

* USE THIS SHEET IN ADDITION TO MIKE BALLMANN' INSTRUCTIONS.

AFTER YOU COMPLETED THIS PROJECT AND YOU HAVE THE TITLE SCREEN, INSERT EXTENDED BASIC AND TYPE SIZE AND HIT RETURN KEY. IF EVERTHING IS OK YOU WILL SEE THE ADDED RAM

11840 BYTES STACK FREE
24488 BYTES OF PROGRAM SPACE FREE

HM 62256LP-12 GOES PIGGY BACK ON TOP OF U610 & U611



BY M.E ADMINA
WMA "BITS/BYES"

BITS, BYTES & PIXELS

Published by Lima OH

99/4A User Group

Material contained herein may be copied by any user group as long as credit is given. Individual and dealer advertisments are published at no charge at the editors' discretion.

ADDRESS- P.O. Box 647
Venedocia Ohio
45894

Published monthly except
July and August

GROUP OFFICERS

President-David Szippl
419-228-7108

Vice Pres-Melvin Nomina
419-692-9564

Treasurer-Leonard Cummings
419-738-3770

Newsletter editor and
Librarian-Charles Good
419-667-3131

BB&P ARTICLE IS "PICK OF THE MONTH"

The Funnelweb 4.0 review and accompanying flow charts by Charles Good, published in the Dec. 87 issue of BB&P was rated "Pick of the Month" by the TI Sig of the Boston Computer Society. They give this designation each month to the best TI newsletter article among all those they receive. BB&P and the Lima User Group are grateful for the honor.

The Boston Computer Society invites everyone to attend the annual TI FAYUH to be held in Lexington MA on Saturday April 2, 1988.

THE LIMA OHIO MULTI USER GROUP CONFERENCE

ATTENTION ALL GREAT LAKES AREA TI USER GROUPS AND DEALERS: The Lima Ohio User Group is organizing a multi TI-User Group conference for Saturday May 21 on the Lima Campus of Ohio State University. User Groups and dealers will be encouraged to attend. The conference will be held in the campus cafeteria, where there are lots of tables and chairs avialable to set up group and dealer displays. We also will have the use of a large conference room for seminars and demonstrations. The Lima User Group will provide monitors and a TI system for use in this conference room.

The best part is that this activity is FREE. There will be NO CHARGE to user groups or dealers who wish to set up booths and/or give demonstrations in the conference room. There will be NO ADMISSION CHARGE to individuals.

This will be a great time for great lakes area user group personalities to meet each other. Groups can look over and hopefully obtain copies of goodies from each others' libraries. We hope to have some interesting demonstrations scheduled for our conference room.

More information will be forthcoming by phone, mail, and in future issues of this newsletter. . Groups and dealers needing tables for displays will need to preregister. For now, keep SATURDAY MAY 21 open for fun and fellowship.